Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02080133.8

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr: Application no.: Demande no: 02080133.8

Anmeldetag: Date of filing: Date de dépôt: 04.12.02

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Optimizing a discovery process using configuration numbers

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F9/40

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SI SK

Optimizing a discovery process using configuration numbers

-- Introduction --

The UPnP 1.0 device architecture [1] consists of six parts: addressing, discovery, description, control, eventing and presentation. In this technical note we focus on the way discovery interacts with description.

Discovery

The discovery process describes how devices that implement UPnP 1.0 control points can discover other devices that implement UPnP 1.0 controlled devices. Basically, a control point can listen to announcement messages from controlled devices. Controlled devices will periodically broadcast these announcements. Furthermore, control points can explicitly broadcast a request for announcements (a so-called M-SEARCH), if they do not want to wait for the periodic refresh. Controlled devices react to search requests by unicasting an announcement to the requesting control point.

The discovery process also describes how control points can discover that specific controlled devices are not longer available. The UPnP 1.0 device architecture describes two mechanisms: first, devices can announce that they will no longer be available, by sending a byebye message. However, there are circumstances in which a device cannot send a byebye message. For example, a device cannot send a byebye message in the event of sudden power loss, or a sudden network disconnection. To cover these events, all device announcements have a time-to-live. When the time-to-live expires, a control point can assume that the controlled device has left the network.

Description

Once a control point has discovered a controlled device, it can proceed to retrieve the device and service descriptions. In general, the discovery process provides a control point with a

rough idea of the capabilities of a controlled device (device type, provided services). The device and service descriptions explicitly and in detail describe the capabilities of the device (icon, friendly name, manufacturer, supported optional features, vendor extensions, allowed parameters, etc.).

Due to their size and complexity, retrieving these descriptions poses quite a burden on the involved devices and the network. The UPnP 1.0 device architecture specifies that a control point can cache these descriptions as long as the corresponding discovery advertisements have not expired. This caching mechanism decreases the load on UPnP devices. However, if due to a temporary disconnect of the network advertisements time out, all control points will need to refresh their cache and download the descriptions. This places a peak load on an UPnP device, just after a time-out.

– Problem definition –

The UPnP 1.0 device architecture describes a two-step mechanism: discovery and description. While these two steps could be combined into a single step, having a two-step mechanism allows for effective caching of static information, which reduces the load on the network and the involved devices. The first step deals with the dynamics of the network: appearing, changing and disappearing devices. The second step provides a detailed view of the capabilities of the device, but is inherently less dynamic due to the size of the involved messages.

However, the caching mechanism as described in the UPnP 1.0 architecture leads to peak loads after temporary network disconnections: cached information is invalidated by a time-out and needs to be refreshed. A new caching rule is needed that avoids these peak loads.

In general, a caching rule for discovery and description is needed that satisfies the following requirements:

- Cache consistency. If the device and service descriptions of a controlled device are cached, and subsequently the controlled device changes its configuration, the cache should as soon as possible be invalidated.

- Large number of cache hits. To minimize the load on the network and on involved devices, device and service descriptions should preferably be served from the cache, instead of being retrieved over the network.

-- Caching using configuration numbers --

We observe that the amount of configurations that a controlled device actually announces to the network can be quite limited. In many cases, there are only two configurations: either the device does not offer any services to the network, or the device offers a fixed set of services to the network.

However, there are those cases where a device can support a number of different configurations, and switch between them. Our proposal is to define a caching mechanism that not only allows caching of the most recent configuration (in UPnP 1.0 device architecture: a set of description files), but also caching of earlier configurations, for rapid lookup. This is enabled by identifying individual configurations with a configuration number. A controlled device transmits its current configuration number in the first step (discovery). This allows control points to detect at the earliest time whether any cached configuration is still current (if a different configuration number is obtained). A control point can decide on the basis of the configuration number whether the information is stored in the local cache, or whether it needs to retrieve the information in step two (description).

We propose the following rules for configuration numbers:

- Configuration number 0 (or any other constant) specifies "do not cache". This rule allows controlled devices that do not know their own configuration to "opt out".
- If a controlled device sends out two messages with the same configuration number K<>0, this ensures that the device configuration is the same at the moments that these messages were sent.

These liberal rules allow devices to "opt out", and also allow devices to give different numbers to the same configuration. This last option can be useful since the internal state of a device consists of more than the configuration information of the device and service description. This might make it difficult to detect that two internal configurations map to the

same set of description files. The liberal rule enables simple device implementations that fail to detect such similarity.

Having such a configuration number allows control points to maintain an extensive cache, not only for the current configuration, but also for past configurations, that could be reused in the future.

To minimize required standardization and to simplify implementations, each controlled device can independently assign numbers to its own configurations. For example, it can maintain a lookup table, have an internal state-transition-machine, or hash the set of description files. This allows caching per device. If cache size becomes a limiting factor, configuration numbers can be standardized for each device-type. This allows caching per device-type. However, we feel that with current advances in storage capacity such additional standardization is not worth the effort.

When applied to UPnP, our proposal is to include a "configuration number" in ssdp:alive messages. If an UPnP device sends out two ssdp:alive messages with the same configuration number, this ensures that the device configuration is the same (same root device, embedded devices, services). This allows control points to (indefinitely) maintain a list of triplets: (device ID, configuration number, descriptions). Such an extended cache eliminates the need to download the same description twice.

Note that the proposal can be used in more advanced systems than in the UPnP 1.0 device architecture. UPnP 1.0 does not allow that a device changes its configuration while in operation. A device has to first leave the network (by sending byebye messages), and then reappear, announcing a new configuration. This can lead to inconsistent caches, for example when a control point misses the byebye messages, it can be unaware of the configuration change. Using the proposed configuration number, devices can change their configuration while continuing to offer services to the network without a full interruption. As an extension a "in transition" bit can be added to the discovery messages, that warns control points that a configuration change is about to happen.

The advantage of this proposal is that it reduces peak loads on UPnP devices during startup / network hiccups. Only if a control point receives an announcement of an unknown configuration is downloading required.

Example

The example consists of a UPnP controlled device A and a UPnP control point B. Device A has a certain configuration C1, in which a service S supports optional features s1, s2, and s3. Device A has a second configuration C2, in which a service S supports optional features s1 and s4.

In the following, we show which messages are no longer required when configuration numbers are added to announcement-messages. We describe the situation of a network hiccup and a continuous switching between two configurations.

step 0: initial state.

Device A and B are unaware of each other. Device A has current configuration C1.

step 1: first discovery.

Device A sends one or more announcement messages to device B. If configuration numbers are used, number #C1 is sent as well.

Device B receives the announcements. It does not have the descriptions in its cache, and downloads and caches them.

step 2: network hiccup and subsequent discovery.

The cache of device B is timed-out.

Subsequently, device B receives an announcement message of device A. If configuration numbers are used, number #C1 is contained in the announcement.

If no configuration numbers are used, device B downloads the descriptions again. Using configuration numbers, device B discovers that the configuration is unchanged: #C1. Its configuration is still in the cache

step 3: switching to configuration C2

Device A changes configuration to C2 and sends byebye messages.

These are received by device B. If no configuration numbers are used, device B will clear its cache.

Subsequently, device A sends new announcements. If configuration numbers are used, number #C2 is sent as well.

These are received by device B. If device B does not use configuration numbers it has an empty cache and it will download the descriptions. If device B uses configuration numbers, it will check whether a configuration (A, #C2) is cached. this will not be the case, and B downloads the descriptions.

step 4: switching to configuration C1

Device A changes configuration to C1 and sends byebye messages.
These are received by device B. If no configuration numbers are used, device B will clear its cache.
Subsequently, device A sends new announcements. If configuration numbers are used, number #C1 is sent as well.
These are received by device B. If device B does not use configuration numbers it has an empty cache and it will download the descriptions. If device B uses configuration numbers, it discovers that (A, #C1) is already in the cache and does not download anything.

step 5: switching to configuration C2

Device A changes configuration to C2 and sends byebye messages.
These are received by device B. If no configuration numbers are used, device B will clear its cache.

Subsequently, device A sends new announcements. If configuration numbers are used, number #C2 is sent as well.
These are received by device B. If device B does not use configuration numbers it has an empty cache and it will download the descriptions. If device B uses configuration numbers, it discovers that (A, #C2) is already in the cache and does not download anything.

step 6: go to step 4.

-- Conclusions --

The two-phase discovery and description process that is described in the UPnP 1.0 device architecture provides a good engineering trade-off between two requirements:

1. The need for fast, dynamic updates (which requires messages to be sent at short intervals, thus messages need to be short to limit overhead)
2. The need for detailed configuration descriptions (which requires long messages)

The trade-off allows fast discovery of any changes in the network, and in a separate, slightly slower step, the retrieval of the entire set of descriptions. Furthermore, a caching mechanism is defined.

In this document we have shown how the caching mechanism can be improved, based on the fact that most devices have only a limited number of different configurations. Specifically, by assigning a configuration number to each different configuration, and transmitting that already in the messages that are used for discovery, caching of all configurations (instead of just the most recent) becomes possible.

This advantage of this approach is that control points only need to download detailed descriptions of configurations they have never downloaded before. This reduces peak loads on devices during startup, configuration transitions and network hiccups.

-- Relation to WSDL / webservices --

Discovery and description of webservices, for example by means of the Web Services Description Language (WSDL), can also be two separate steps, closely related to how it is described here.

Sometimes, the exact configuration (supported capabilities) are not put in description files, but the supported capabilities can be retrieved using a "getCapabilities()" action. The validity of such a getCapabilities() action also depends on whether the configuration of the services changes. To avoid repeatedly invoking getCapabilities(), configuration numbers can be used, using the following rule for configuration numbers:

- if the configuration number is the same in two instances, any call getCapabilities() (or similar) action will return the same result.

-- Flowcharts --

Fig. 1 shows the processing steps in a (controlled) device. Starting at step 100, the device will select an initial configuration of its network Services, at step 102. The device will then assign a number N to this configuration, in step 104, according to our rules for configuration numbers. From then on, it will go to a state, in step 106, from where the device either reacts to searches and in the search response includes N, in step 108, or, at specific times (e.g. periodically), sends out advertisements that include N, in step 110. Following step 108 or step 110, the device returns to step 106. Also, following step 106, the device can decide to reconfigure, choosing a new configuration, by continuing with step 2.

Fig. 2 shows the processing steps in a control point. Starting at step 200, the control point goes to a state, in step 202, from where it can send search requests, in step 204. Also in step 202, the control point listens to both incoming advertisements and search responses (which are actually the same kind of messages). When they arrive, in step 206, the control point handles them. It checks, in step 208, whether it already cached the configuration belonging to the concerned device and configuration number. If so, the control point does not have to do anything else and it returns to the listening state, in step 202. If not so, the control point retrieves the new configuration description, in step 210 (note: it can delay this if no application is currently looking for additional devices).

Out of scope:

- disappearing devices (does not influence caching),
- control point clearing the cache to cache limit size
  - cache clearing strategy typically keeps currently-in-use device configurations
  - traditional least-frequently-used strategies on cached descriptions apply.

— Summary —

Adding a configuration number to enable effective caching in control points.

The UPnP DA 1.0 architecture specifies that control points can retrieve device and service descriptions at any time. It further specifies that control points can cache these descriptions as long as discovery advertisements from a device have not expired. This caching mechanism decreases the load on UPnP devices. However, if due to the unreliable network an advertisement is timed out, all control points will need to refresh their cache and download the descriptions. This places a peak load on an UPnP device, just after a time-out.

Our proposal is to include a "configuration number" in ssdp:alive messages. If an UPnP device sends out two ssdp:alive messages with the same configuration number, this ensures that the device configuration is the same (same root device, embedded devices, services). This allows control points to (indefinitely) maintain a configuration number to description cache that eliminates superfluous requests for description pages.

The advantage of this proposal is that it reduces peak loads on UPnP devices during startup or network hiccups.

-- References --

[1] Universal Plug and Play Device Architecture, Microsoft Corporation, www.upnp.org, 2000.

CLAIMS:

1. A method and arrangement substantially as hereinbefore described.

ABSTRACT:

Adding a configuration number to enable effective caching in control points. The UPnP DA 1.0 architecture specifies that control points can retrieve device and service descriptions at any time. It further specifies that control points can cache these descriptions as long as discovery advertisements from a device have not expired. This caching mechanism decreases the load on UPnP devices. However, if due to the unreliable network an advertisement is timed out, all control points will need to refresh their cache and download the descriptions. This places a peak load on an UPnP device, just after a time-out. Our proposal is to include a "configuration number" in ssdp:alive messages. If an UPnP device sends out two ssdp:alive messages with the same configuration number, this ensures that the device configuration is the same (same root device, embedded devices, services). This allows control points to (indefinitely) maintain a configuration number to description cache that eliminates superfluous requests for description pages. The advantage of this proposal is that it reduces peak loads on UPnP devices during startup or network hiccups.

Fig. 1

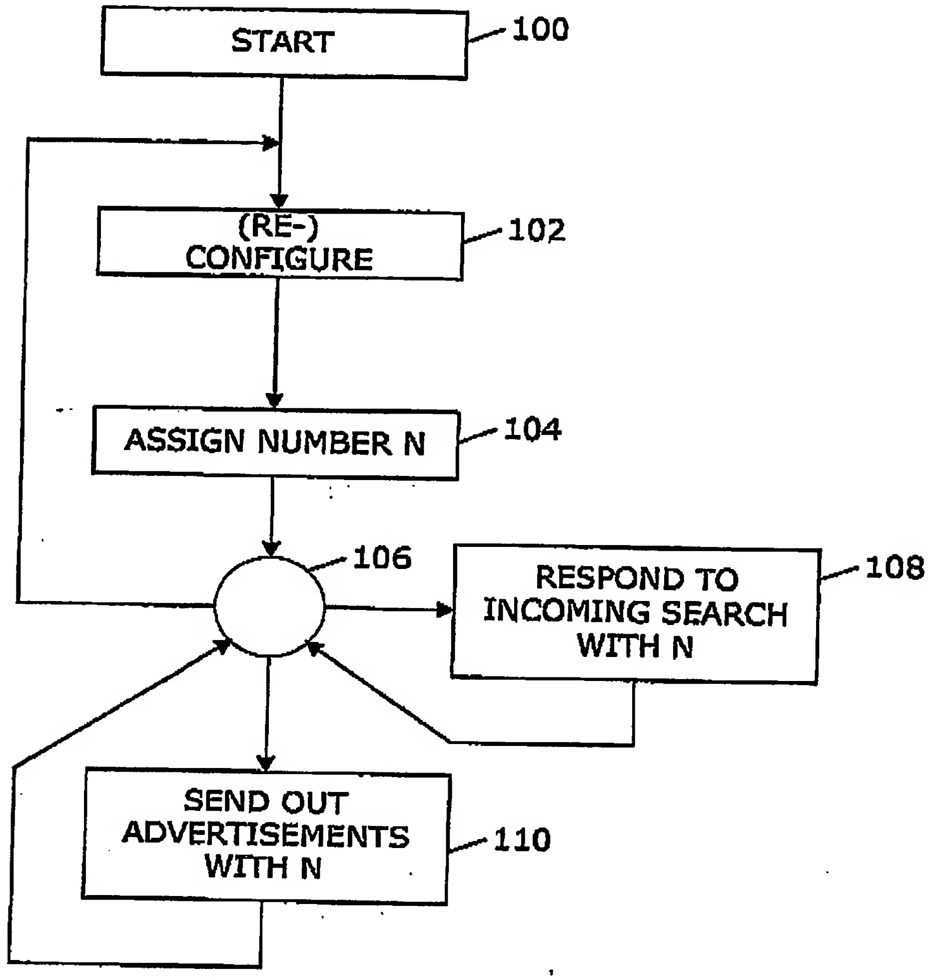
START
100
(RE-)
CONFIGURE
102
ASSIGN NUMBER N
104
106
RESPOND TO
INCOMING SEARCH
WITH N
108
SEND OUT
ADVERTISEMENTS
WITH N
110

FIG. 1

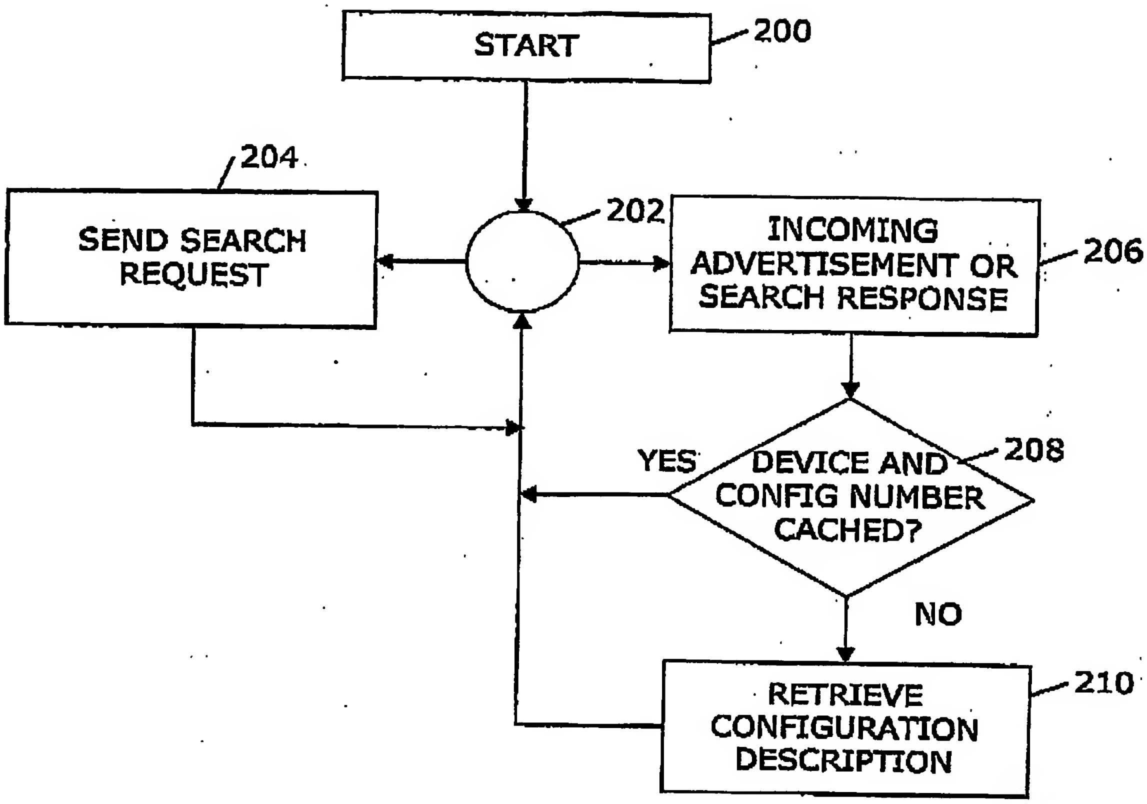
START
200
204
SEND SEARCH REQUEST
202
INCOMING ADVERTISEMENT OR SEARCH RESPONSE
206
YES
DEVICE AND CONFIG NUMBER CACHED?
208
NO
RETRIEVE CONFIGURATION DESCRIPTION
210

FIG. 2